GAN : Generative Adversarial Nets

Abstract

-2개의 모델(G,D) 학습시킨다

-G: gernerative model로, 진짜와 비슷한 이미지를 만들어 낸다.

-D : Discriminateive model로, 이미지가, real이미지인지, fake이미지인지 판별하는 역할을 한다.

D(discriminative model은, real,fake여부를 판별하는 것이므로, G(generative model은 D(discriminative model)model이 만들어낸 이미지(G)를 진짜이미지로 판별 할 수 있는 확률을 최대로 되게 학습을 시켜야 한다.

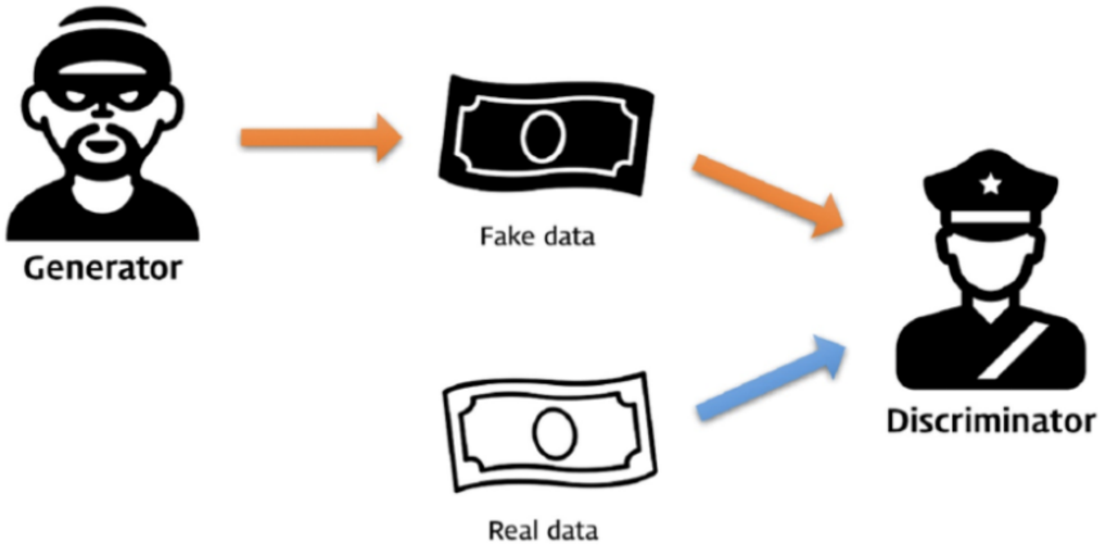
-G가 충분히 학습하게 된다면, D는 real data인지, G가 만들어낸 data인지 판별할 수 있는 확률은(1/2)가 된다. 즉, D는 제대로 된 판별을 하지 못한다.

-G,D는 다중퍼셉트론으로 정의된다면, 역전파를 통해 학습된다

→ 본 논문은, 생성 모델(G)와 판별 모델(D) 두 가지 모델을 학습하며, G는 real data분포가 되도록 학습하고, 이와 비슷한 데이터를 생성하려고 한다. 반면, D는 real data와 G가 생성한 fake data를 올바르게 판별하려는 게임형식으로 진행이 된다.

Introduction

- Deep generative model은 많은 다루기 힘든 확률적 계산때문에, generative context의 이점을 활용하기 어려웠다. 본 논문에서는 이러한 문제점들을 피하기 위한, generative model을 제안한다.
- discriminative model은 sample이 model distribution인지, data distribution인지 잘 판별하는 방향으로 학습한다. (경찰 역할)
- generative model은 real data와 유사한 데이터 분포를 가질 수 있는 방향으로 학습한다.(위조범 역할)
- 이러한 학습은, real data와 fake data를 잘 구별하지 못할때까지 진행되며, 모식도는 다음과 같다.



- G와 D에 대한 학습이 충분히 진행되면, 경찰(D)는 fake를 판별 할 수 있는 확률이 (1/2)로 수렴하게 된다 (진짜인지 가짜인지 찍는 느낌)

→ GAN의 concept은 각각의 역할을 가진 두 모델을 적대적으로 학습시키며, '진짜같은 가짜'를 생성해내는 능력을 키워주는 것이다.

Adversarial nets

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

D(x) : x가 pg(fake)에서 나온 것이 아닌 data(real)에서 나왔다고 판별할 확률(0~1) → D(x)~1 : real data라고 판별, D(x)~0 : fake data라고 판별

p data: real

p g: fake

→ G를 학습시킬수록, D(G(z))~1이 되고, D를 학습시킬수록 D(G(z))~0이 된다.(상대적인 관계)

1. **D 입장에서 보기**

첫 항에서, real data를 real data라고 잘 판단하면, D(x)~1이므로, log(1)=0 즉, 첫 항은 0이 되어 사라진다.

둘째항에서, fake data를 fake data라고 잘 판단하면, D(G(z))~0이므로, log(1-0) 즉, 둘째항도 0이 되어 사라진다.

다시 말해, D의 입장에서는 minmaxV(D,G)~0이 되는게 이상적인 학습방향이라고 볼 수 있다. 즉, V(D,G)를 최대값으로 키워야하는게 D의 입장이다.

2. **G의 입장에서 보기**

첫 항은, G가 만들어낸 data가 아닌 real data에 관한 부분이므로, 무시할 수 있다.

둘째항은, G는 D가 진짜(D(G(z))~1))라고 판별할 수 있는 방향으로 학습을 시켜야 하므로, D(G(z))~1이 나오는 것이 G의 입장에서는 best case라고 볼 수 있다.

즉 G의 입장에서는 minmaxV(D,G)가 log(1-D(G(z)) ~ log(1-1) ~ -Infinity, 최소값으로 나오도록 학습시키는게 이상적이다.

→ **G의 입장에서는 V(D,G)가 min이 되도록, D의 입장에서는 V(D,G)가 max가 되도록 학습시키는 방향이 이상적**이므로, 이는 논문에서 minmax-game이라고 설명하고 있다.

- 학습시킬때, inner loop에서 D를 최적화하는 것은 많은 계산들이 필요하므로, finite dataset에서 overfitting이 발생한다.

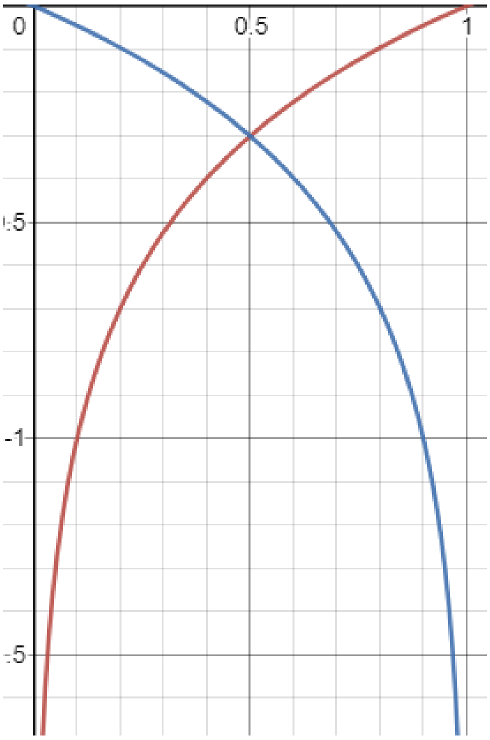
→ 이의 대안으로, D를 k step optimize할때, G는 1 step optimize시켜준다. 즉 ,D와 G가 학습되는 balance를 맞춰준다(?)

- 학습 초기(G의 성능이 좋지 않을 때)에는 D가 real,fake를 잘 구별한다.

→ log(1-D(G(z))가 금방 포화되고, 이를 최소화 하는 것보다는 log(D(G(z))릴 최대화 시키는 것이 학습 초기에서 더 좋다고 합니다.

→G의 성능이 좋지 않으면, D(G(z))~0이므로, log(1-D(G(z)) ~ -INF가 되는데, 이때 gradient가 너무 작게 나오므로 학습이 느리다.

→ 즉, 아래 그래프에서 파란색(log(1-x)를 최소화하기보단, log(x))를 최대화하는 방향으로 학습시키는 것이 좋다(?)



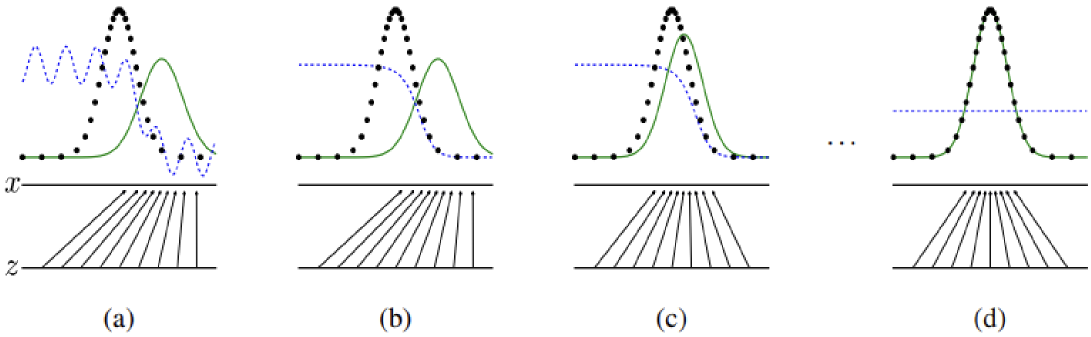


Figure 1: Generative adversarial nets are trained by simultaneously updating the **d**iscriminative distribution (D, blue, dashed line) so that it discriminates between samples from the data generating distribution (black, dotted line) $p_{\boldsymbol{x}}$ from those of the **g**enerative distribution p_g (G) (green, solid line). The lower horizontal line is the domain from which $\boldsymbol{z}$ is sampled, in this case uniformly. The horizontal line above is part of the domain of $\boldsymbol{x}$. The upward arrows show how the mapping $\boldsymbol{x} = G(\boldsymbol{z})$ imposes the non-uniform distribution p_g on transformed samples. G contracts in regions of high density and expands in regions of low density of p_g. (a) Consider an adversarial pair near convergence: p_g is similar to p_{data} and D is a partially accurate classifier. (b) In the inner loop of the algorithm D is trained to discriminate samples from data, converging to $D^*(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})+p_g(\boldsymbol{x})}$. (c) After an update to G, gradient of D has guided $G(\boldsymbol{z})$ to flow to regions that are more likely to be classified as data. (d) After several steps of training, if G and D have enough capacity, they will reach a point at which both cannot improve because $p_g = p_{\text{data}}$. The discriminator is unable to differentiate between the two distributions, i.e. $D(\boldsymbol{x}) = \frac{1}{2}$.

파랑 점선: D(discriminator)

검정 점선: real dataset distribution

초록 실선: generator가 생성한 dataset distribution(fake)

z : real data(x)기준으로 g가 생성한, sampled data(fake)가 어디로 mapping 되어져 있는지

(a) 학습 초기 : real와 fake의 분포가 전혀 다르며, D의 성능도 그닥 좋지 않다.

(b): (a)에서 D가 조금 학습되면, D가 어느정도 real과 fake를 판별할 수 있데 된다. 이때 D는 다음처럼 수렴한다.

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

(c) : G가 조금 학습 되면, generator가 만들어내는 dataset의 분포가, real data의 분포와 가까워진다.

(d): 학습을 충분히 하였을 때, G의 data 분포가 real 분포와 같게 되어(pg~pdata) D는 올바르게 real인지 fake인지 판별할 수 없게 된다.(1/2)

여기서 D가 1/2로 수렴하게 되는 이유는, 직관적인 관점에서는 data가 real인지 fake인지 구분을 하지 못하므로, 찍게되어 1/2로 나오는 것으로 이해해도 되고,

D를 학습시킬때는 D(G(z))~0이 되도록 학습시키고,

G를 학습시킬때는 D(G(z))~1이 되도록 학습시키므로, 이 둘을 충분히 학습시켰을때는, 0과 1사이인 1/2로 수렴하게 된다고 이해해도 되는 것 같습니다.

Theoretical Results

minmax problem에서 한쪽 모델의 성능을 높이면 나머지 한 쪽은 성능이 떨어지므로, 모두를 만족시키는 평형점을 찾아야합니다.

→ 이의 해결방안으로, 한 쪽은 상수로 고정시키고, 다른 변수에 대해서 문제를 푸는 방법을 제시하고 있습니다.

ex)

maxV(D,G')에서 G'을 상수로 고정하고, D와 관련하여 문제를 푼다. 이때 구한 D값을 D'이라 한다면, 이제는 minV(D',G)를 통해서 G값을 구할 수 있습니다.

→ 정의한 문제가 실제로 정답이 있는지(existence), 해가 존재한다면 유일해인지(uniqueness), 제안한 방법이 실제로 원하는 해를 찾을 수 있는지(convergence)를 확인해야 합니다.

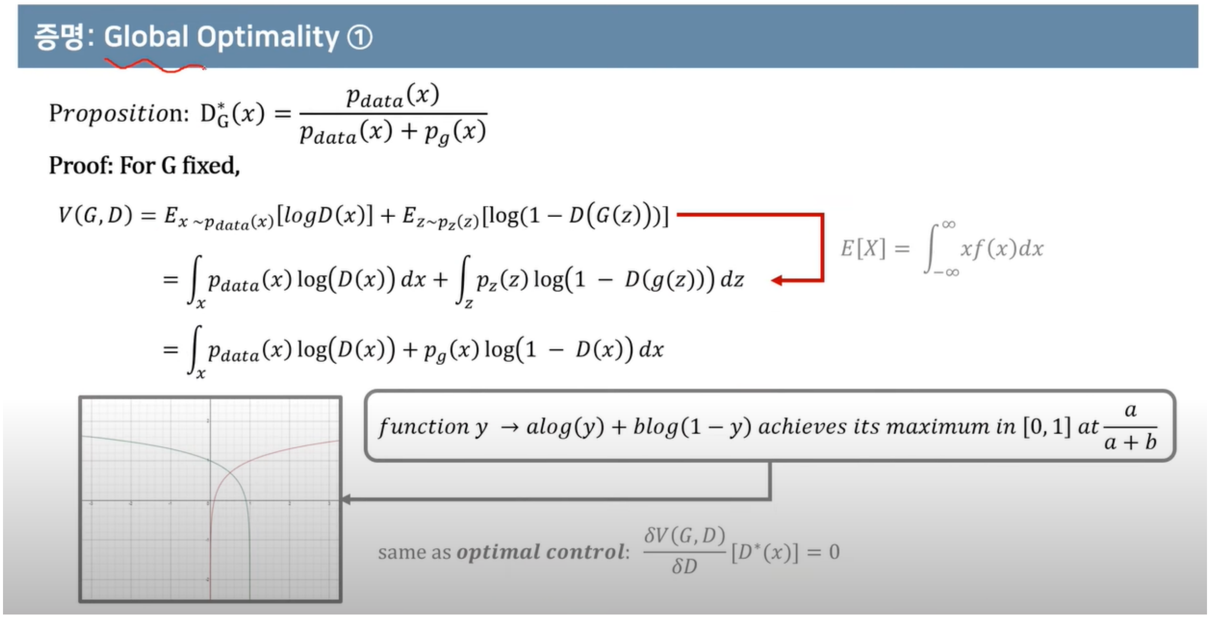
1. Global Optimality of pg=pdata

최적의 판별자 D를 고려한다

Proposition 1: G가 고정 된 상태에서 최적의 D를 구하면, 아래와 같습니다.

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

Proof :



출처 : (https://www.youtube.com/watch?v=AVvlDmhHgC4)

D입장에서는 어떠한 G가 오든, V(G,D)를 maximize 시키는 것이 목표입니다.

E는 기대값(평균값)이므로, 위와 같이 적분식으로 나타낼 수 있습니다.

For any $(a, b) \in \mathbb{R}^2 \setminus \{0, 0\}$, the function $y \to a\log(y) + b\log(1-y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$. The discriminator does not need to be defined outside of $Supp(p_{\text{data}}) \cup Supp(p_g)$,

이 때, (a,b)의 실수 순서쌍(a,b)를 보면 (a=Pdata , b=Pg)

alog(y)+blog(1-y)로 나타낼 수 있고, 이의 최댓값은 이 식을 미분하면 a/y-b(1-y)=0이 되고, 이는 a(1-y)-by=0으로 정리할 수 있으며, y(a+b)=a이므로, 최종적으로 V(G,D)의 최댓값은 y=a/(a+b) 일 때 이므로,

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

최적의 D는 다음과 같음을 알 수 있습니다.

이를 다시 minmax-problem에 대입하게 되면,

증명: Global Optimality ②

$Proposition$: $Global\ optimum\ point\ is\ \boldsymbol{p_g} = \boldsymbol{p_{data}}$

Proof:

$$C(G) = \max_D V(G,D) = E_{x\sim p_{data}(x)}[\log D^*(x)] + E_{z\sim p_z(z)}[\log(1 - D^*(G(z)))]$$

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

$$= E_{x\sim p_{data}(x)}\left[\log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}\right] + E_{x\sim p_g(x)}\left[\log \frac{p_g(x)}{p_{data}(x) + p_g(x)}\right]$$

removed when $p_g = p_{data}$

$$KL(p_{data}||p_g) = \int_{-\infty}^{\infty} p_{data}(x)\log\left(\frac{p_{data}(x)}{p_g(x)}\right)dx$$

$$= \boxed{E_{x\sim p_{data}(x)}\left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_g(x)}\right] + E_{x\sim p_g(x)}\left[\log \frac{2 * p_g(x)}{p_{data}(x) + p_g(x)}\right]} - \log(4)$$

$$= KL\left(p_{data}\middle|\middle|\frac{p_{data}(x) + p_g(x)}{2}\right) + KL\left(p_g\middle|\middle|\frac{p_{data}(x) + p_g(x)}{2}\right) - \log(4)$$

$$= 2 * JSD(p_{data}||p_g) - \log(4)$$

$$JSD(p||q) = \frac{1}{2}KL\left(p\middle|\middle|\frac{p+q}{2}\right) + \frac{1}{2}KL\left(q\middle|\middle|\frac{p+q}{2}\right)$$

3번째 줄과 같이 나오게 되고, 각 log 항 내부에 2를 곱하고, 밖으로 -log(4)로 빼낸 형태로 바꿔줄 수 있다.

이때 KL(p||q)는 KL-divergence (Kullback - Leibler divergence)로, p라는 분포가 있을 때, q와 p가 얼마나 다른지, 즉, 확률 분포의 차이를 계산하는데 사용된다.

KL-divergence는 다음과 같이 정의되며, 이의 값이 작을수록, 두 분포는 유사합니다.

$$KL(p \parallel q) = \begin{cases} \sum_i p_i \log \frac{p_i}{q_i} \ \text{또는} \ -\sum_i p_i \log \frac{q_i}{p_i} \quad (\text{이산형}) \\ \int p(x) \log \frac{p(x)}{q(x)} dx \ \text{또는} \ -\int p(x) \log \frac{q(x)}{p(x)} dx \quad (\text{연속형}) \end{cases}$$

따라서, KL-divergence를 사용하여, 4번째 줄과 같이 나타낼 수 있으나,

KL-divergence는 거리의 대칭성이 성립하지 않아, 대칭성을 갖도록 변형시킨 JSD를 사용합니다.

JSD는 Jensen-Shannon divergence의 약자로, 다음과 같이 나타냅니다.

$$JSD(p\|q) = \frac{1}{2}KL(p\|M) + \frac{1}{2}KL(q\|M)$$
$$where, M = \frac{1}{2}(p+q)$$

이로 인해, KL이 혼합된 4번째 줄의 식에서, 마지막번째 줄의 식으로 치환이 가능합니다.

JSD는 두 확률분포가 같을 때만 0이고, 나머지에서는 0보다 큰 양수를 가지므로,

즉 Pg=Pdata일때, C(G) = -log(4)라는 global minimum을 갖습니다.

또 Pg=Pdata라는 의미는 G가 생성해낸 데이터가 실데이터와 같은 분포를 보인다는 의미이므로, 위의 Figure(1)에서 (d)에 해당합니다. 이는, D(x)=1/2와 같은 의미이므로, D(x)=1/2를, minmax-problem식에 대입하면 똑같이 C(g)=-log(4)라는 값을 얻을 수 있습니다.

즉 위의 내용은 Theorem 1에서 다시 볼 수 있습니다.

Theorem 1

C(G)는 pg=pdata일때 global minimum을 갖고, 그때의 C(G)값은 -log(4)이다.

이에 대한 증명은 바로 위에서 말한, C(G)에 D(x)=1/2를 대입하였을 때가,

$$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right)$$

아래와 같은식과 동일할 때, 즉 JSD가 0(두 확률 분포가 같을 때만 0이고, 나머지는 0보다 큰 양수)일때와 동일합니다.

Algorithm 1

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.

  • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.

  • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

  end for

  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.

  • Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Algorithm 1은 위에서 말한,

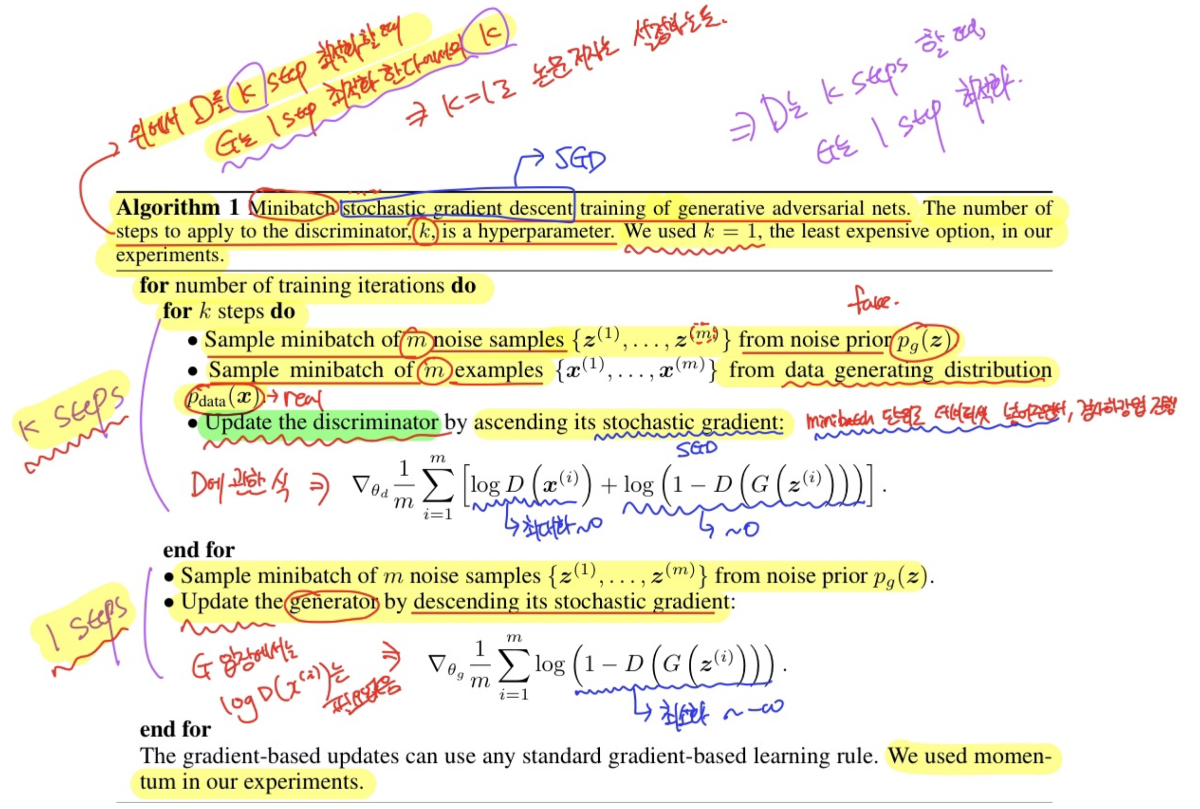
- **학습시킬때, inner loop에서 D를 최적화하는 것은 많은 계산들이 필요하므로, finite dataset에서 overfitting이 발생한다.**

  → 이의 대안으로, D를 k step optimize할때, G는 1 step optimize시켜준다. 즉 ,D와 G가 학습되는 balance를 맞춰준다.

의 동작 과정을 보여줍니다.

본 논문의 실험에서는 k=1을 사용하였고, SGD를 사용하였습니다. 방법은 기존 SGD방식과 같은 것 같습니다.

말로 설명하기가 쉽지 않아, 필기한 내용을 첨부하겠습니다.



Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**
  for k steps **do**
  • Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.
  • Sample minibatch of m examples $\{x^{(1)}, \dots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
  • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right].$$

  end for
  • Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.
  • Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right).$$

end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

위의 알고리즘을 보면, 본 논문의 저자가 말한것처럼, **D를 k step optimize할때, G는 1 step optimize 시켜주는 것을 볼 수 있습니다.**

Convergence of Algorithm 1

#추가 예정...(잘 이해가 안되네요...)

minmax 에서 global optim을 가지는데, 이때는 pg=pdata 일때일뿐이다 라는 것을 증명하였으므로,

풀고자 하는 알고리즘이 실제로 pg=pdata로 수렴하는지에 대한 증명

“The subderivatives ~ attained”

→ f(pg)=supD fD(pg) → sup을 max로 간단하게 보면, D에 대해서 max한 loss 함수 fD(pg)이다. 이는 pg에 대한 value loss → 이것이 convex 인데, 모든 pg에 대해서 convex를 한다는 것이고, D도 sup D(D이므로, 모든 D에서도 성립하기때문에, alphafD*(pg) (alpha f, alpha f : subderivatives set에 들어간다.

→ subderivatives : f라는 함수의 미분들의 set (여기에는 D가 optimal일때의 미분 set도 alpha fD*(pg) 에포함됨 → D가 포함되어있는 일반적인 함수는 fD(pg)인데, 이 함수가 pg에 대해서 convex라는 얘기는, C(G)가 convex하다는 것과 같음. → 우리가 풀고자 하는 문제가 convex 문제라면 minimize는 쉬움(global optim 존재하기때문에) → 우리가 풀고자 하는 문제가 convex이기 때문에, global optimal에 항상 간다.(gradient descent해서 가게 되면 global optimal 도달하는게 보장되있기 때문에) → convex한 이유는 C(G)에서 JSD는 positive한 함수 (pg=pdata를 제외한 것은 모두 양수)이기 때문에 convex하다고 할 수 있음

fD(pg)는 다양한 case의 D에서 maximum에 해당하는 것?

결론: convex하므로, global minima에 도달할 수 있는 것이 보장되어져 있고, global minima근처에서 미분을 이용한 경사하강법을 진행하면 global minima에 도달 할 수 있다. 그래서 algorithm에서 SGD를 사용한다.

Experiment

- G의 activation 함수로는 rectifier linear, sigmoid를 혼합하여 사용했다고 합니다.
- D는 maxout activation을 사용하였고, 학습시킬때, dropout을 사용했다고 합니다.
- G가 생성해낸 sample이 기존 방법들로 만든 sample보다 좋다고 주장할 수는 없지만, adversarial net은 잠재력이 있다.

Advantages and disadvantages

Advantages

- Markov chains은필요없고 역전파 알고리즘만이 사용된다.
- Inference가 학습중에 필요없다.
- 많은 method 들이 본 model에서 사용할 수 있다.
- G를 통해 만들어진 이미지를 매우 정확하게 나타낼 수 있다.

Disadvantages

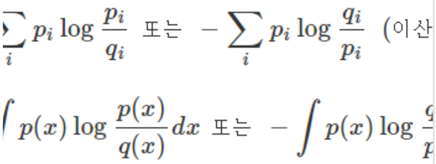
- Pg(x)를 정확하게 나타낼 수 없다.
- D는 학습 도중에 G와 잘 synchronized되어야 한다.
- G는 D를 업데이트 하기 전에, 많이 train되면 안된다

Reference

KL Divergence(쿨백 라이블러 발산), Jensen-Shannon divergence

Kullback-Leibler divergence 은 두 확률간의 분포의 차이정도를 정량적으로 측정하는 방법이다. 어떤 이상적인 분포에 대해, 그 분포를 근사하는 다른 분포를 사용해 샘플링을 한다면 발생할 수 있는 정보 엔트로피차이를 계산한다. 상대 엔트로피

https://dodonam.tistory.com/80



(GAN)Generative Adversarial Nets 논문 리뷰

(ResNet)Deep Residual Learning for Image Recognition 논문 리뷰

https://tobigs.gitbook.io/tobigs/deep-learning/computer-vision/gan-generative-adversarial-network

GAN: Generative Adversarial Networks (꼼꼼한 딥러닝 논문 리뷰와 코드 실습)

생성 모델(Generative Model)은 실제로는 존재하지 않지만, 있을법한 데이터를 만들어 내는 모델을 의미합니다. 오늘은 현대 딥러닝 기반 생성 모델에 큰 영향을 끼친 논문인 GAN(NIPS 2014)을 소개합니다. GAN은 최근까지 이미지 도메인에서의 많은 발전이 이루어져...

https://www.youtube.com/watch?v=AVvlDmhHgC4